

实验 06：系统监控与安全管理

一、实验目的

- 1、掌握系统性能监控；
- 2、掌握系统可视化监控；
- 3、掌握使用 SELinux 提升系统内核安全性；
- 4、掌握使用 Firewalld 提升系统的安全性。

二、实验学时

2 学时

三、实验类型

设计研究

四、实验需求

1、硬件

每人配备计算机 1 台。

2、软件

安装 VMware WorkStation Pro 或 Oracle VM VirtualBox 软件，安装 Mobaxterm 软件。

3、网络

本地主机与虚拟机能够访问互联网，虚拟机网络不使用 DHCP 服务。

4、工具

无。

五、实验任务

- 1、完成系统性能监控；
- 2、完成使用 Linux-dash 实现系统可视化监控；
- 3、完成使用 SELinux 提升系统内核的安全性；
- 4、完成使用防火墙（Firewalld）提升系统的安全性。

六、实验环境

- 1、本实验需要 VM 1 台。
- 2、本实验 VM 配置信息如下表所示。

虚拟机配置	操作系统配置
虚拟机名称：VM-Lads-06-Task-01-172.31.0.61 内存：1GB CPU：1 颗，1 核心 虚拟磁盘：20GB 网卡：1 块	主机名：Lab-06-Task-01 IP 地址：172.31.0.61 子网掩码：255.255.255.0 网关：172.31.0.254 DNS：172.31.0.254

- 3、本实验拓扑图。
无。
- 4、本实验操作演示视频。
本实验操作演示视频为视频集的第 6 集：<https://www.bilibili.com/video/BV1b1421t7aa?p=6>

七、实验内容及步骤

1、使用命令工具监控系统性能

1.1 查看 CPU 信息

- (1) 使用 lscpu 命令工具查看 CPU 信息。

Shell

```
1 # 使用lscpu显示CPU详细信息
2 [root@Lab-06-Task-01 ~]# lscpu
3 # 以扩展可读的格式显示CPU信息
4 [root@Lab-06-Task-01 ~]# lscpu -e
5 # 显示CPU指定列的信息
6 [root@Lab-06-Task-01 ~]# lscpu -e=CPU
7 # 以可解析的格式显示CPU信息
8 [root@Lab-06-Task-01 ~]# lscpu -p
9 # 显示online CPU信息
10 [root@Lab-06-Task-01 ~]# lscpu -bp
```

(2) 使用 cat /proc/cpuinfo 命令工具查看 CPU 信息。

Shell

```
1 # 使用cat /proc/cpuinfo显示CPU详细信息
2 [root@Lab-06-Task-01 ~]# cat /proc/cpuinfo
3 # 查看CPU型号
4 [root@Lab-06-Task-01 ~]# cat /proc/cpuinfo | grep name | cut -f2 -d: |
  uniq -c
5 # 查看物理CPU个数
6 [root@Lab-06-Task-01 ~]# cat /proc/cpuinfo | grep 'physical id' | sort
  | uniq | wc -l
7 # 查看CPU的总线程数量
8 [root@Lab-06-Task-01 ~]# cat /proc/cpuinfo | grep "processor" | wc -l
```

(3) 使用 mpstat 命令工具查看 CPU 信息。

Shell

```
1 # 使用mpstat命令工具查看CPU信息
2 # 查看多核CPU核心的当前运行状况信息，每两秒更新一次
3 [root@Lab-06-Task-01 ~]# yum install -y sysstat
4 [root@Lab-06-Task-01 ~]# mpstat -P ALL 2
5 # 查看多核CPU核心的当前运行状况信息，每五秒更新一次，采样两次
6 [root@Lab-06-Task-01 ~]# mpstat -P ALL 5 2
```

1.2 查看磁盘信息

(1) 使用 df 命令工具查看磁盘信息。

Shell

```
1 # 使用df命令工具查看磁盘信息
2 [root@Lab-06-Task-01 ~]# df
3 # 使用df -i以inode模式来显示磁盘使用情况
4 [root@Lab-06-Task-01 ~]# df -i
5 # 使用df -T显示文件系统类型
6 [root@Lab-06-Task-01 ~]# df -T
7 # 使用df -h与更易读的方式显示目前磁盘空间和使用情况
8 [root@Lab-06-Task-01 ~]# df -h
9 # 使用df -k以单位显示磁盘的使用情况
10 [root@Lab-06-Task-01 ~]# df -k
11 # 使用df -l显示本地的分区的磁盘空间使用率
12 [root@Lab-06-Task-01 ~]# df -l
13 # 使用df -a显示各文件系统的使用情况
14 [root@Lab-06-Task-01 ~]# df -a
15 # 使用df -ia显示各文件系统的i节点的使用情况
16 [root@Lab-06-Task-01 ~]# df -ia
```

(2) 使用 fdisk 命令工具查看磁盘信息。

Shell

```
1 # 使用fdisk -l显示磁盘当前分区信息
2 [root@Lab-06-Task-01 ~]# fdisk -l
3 # 使用fdisk -lu显示SCSI磁盘的每个分区的情况
4 [root@Lab-06-Task-01 ~]# fdisk -lu
```

1.3 查看系统实时状态

使用 top 命令工具查看系统实时状态。

Shell

```
1 # 使用top命令显示系统运行信息
2 [root@Lab-06-Task-01 ~]# top
3 # 使用top -d设置信息更新时间
4 [root@Lab-06-Task-01 ~]# top -d 3
5 # 使用top -p显示指定进程的信息
6 [root@Lab-06-Task-01 ~]# top -p 192
7 # 使用top -n显示更新3次后推出
8 [root@Lab-06-Task-01 ~]# top -n 3
9 # 使用top -S累计显示进程CPU使用时间
10 [root@Lab-06-Task-01 ~]# top -S
11 # 使用top -H显示进程中线程的详细信息
12 [root@Lab-06-Task-01 ~]# top -H
```

1.4 查看系统性能状态

(1) 使用 htop 命令工具查看系统性能状态。

Shell

```
1 # 使用htop命令工具通过图形操作界面查看系统性能状态
2 [root@Lab-06-Task-01 ~]# htop
```

(2) 使用 sar 命令工具查看系统性能状态。

Shell

```
1 # 使用sar命令工具查看系统性能状态
2 # 使用sar -u查看CPU状态, 每 1s 监控一次, 共监控 3 次
3 [root@Lab-06-Task-01 ~]# sar -u 1 3
4 # 使用sar -r查看内存使用率, 每 1s 监控一次, 共 3 次
5 [root@Lab-06-Task-01 ~]# sar -r 1 3
6 # 使用sar -B查看内存分页情况, 每隔 1s 监控一次, 共 3 次
7 [root@Lab-06-Task-01 ~]# sar -B 1 3
8 # 使用sar -W查看系统交换活动信息, 每隔 1s 监控一次, 共 3 次
9 [root@Lab-06-Task-01 ~]# sar -W 1 3
10 # 使用sar -d查看磁盘使用情况, 每 1s 监控一次, 共 3 次
11 [root@Lab-06-Task-01 ~]# sar -d 1 3 -p
12 # 使用sar -b查看 I/O 和传输率, 每 1s 监控一次, 共 3 次
13 [root@Lab-06-Task-01 ~]# sar -b 1 3
14 # 使用sar -n查看网络情况, 每隔 1s 监控一次, 共 3 次
15 [root@Lab-06-Task-01 ~]# sar -n ALL 1 3
16 # 使用sar -q查看队列的长度与负载的状态, 每隔 1s 监控一次, 共 3 次
17 [root@Lab-06-Task-01 ~]# sar -q 1 3
```

(3) 使用 dstat 命令工具查看系统性能状态。

Shell

```
1 # 使用dstat命令工具查看系统性能状态
2 [root@Lab-06-Task-01 ~]# dstat
3 # 使用dstat -c查看CPU信息, 每隔 1s 监控一次, 共 3 次
4 [root@Lab-06-Task-01 ~]# dstat -c 1 3
5 # 使用dstat -d查看磁盘使用情况, 每隔 1s 监控一次, 共 3 次
6 [root@Lab-06-Task-01 ~]# dstat -d 1 3
7 # 使用dstat -m查看内存使用情况, 每隔 1s 监控一次, 共 3 次
8 [root@Lab-06-Task-01 ~]# dstat -m 1 3
9 # 使用dstat -n查看网络传送信息, 每隔 1s 监控一次, 共 3 次
10 [root@Lab-06-Task-01 ~]# dstat -n 1 3
11 # 使用dstat -p进行进程统计, 每隔 1s 监控一次, 共 3 次
12 [root@Lab-06-Task-01 ~]# dstat -p 1 3
13 # 使用dstat -r进行io请求统计, 每隔 1s 监控一次, 共 3 次
14 [root@Lab-06-Task-01 ~]# dstat -r 1 3
15 # 使用dstat -t进行时间和日期的输出
16 [root@Lab-06-Task-01 ~]# dstat -t
```

2、使用 Linux-Dash 实现系统可视化监控

2.1 部署 Linux-Dash

Shell

```
1 # 使用yum工具安装Apache
2 [root@Lab-06-Task-01 ~]# yum install -y httpd
3 # 使用systemctl start命令启动Apache服务
4 [root@Lab-06-Task-01 ~]# systemctl start httpd
5 [root@Lab-06-Task-01 ~]# systemctl enable --now httpd
6 [root@Lab-06-Task-01 ~]# yum install -y python php php-fpm
7 # 在/var/www/html目录下，下载linux-dash的源码包
8 [root@Lab-06-Task-01 ~]# cd /var/www/html
9 [root@Lab-06-Task-01 html]# git clone https://github.com/afaqurk/linux-dash.git
10
```

2.2 配置安全策略

Shell

```
1 # 配置防火墙，允许80/tcp端口访问
2 [root@Lab-06-Task-01 ~]# firewall-cmd --zone=public --add-service=http --permanent
3 [root@Lab-06-Task-01 ~]# firewall-cmd --reload
4 [root@Lab-06-Task-01 ~]# firewall-cmd --list-all
5 # 配置SELinux
6 [root@Lab-06-Task-01 ~]# setenforce 0
7 # 修改SELinux配置文件
8 [root@Lab-06-Task-01 ~]# vi /etc/selinux/config
9 ----- config -----
10 SELINUX=permissive
11 -----
12 [root@Lab-06-Task-01 ~]# reboot
```

2.3 访问测试

Shell

```
1 # 进行访问安全配置
2 # 修改httpd的配置文件，使用httpd启动方式进行登录保护
3 [root@Lab-06-Task-01 ~]# vi /etc/httpd/conf.d/dash.conf
4 ----- dash.conf -----
5 <Directory /var/www/html/linux-dash/app>
6     Options FollowSymLinks
7     AllowOverride All
8     Order allow,deny
9     allow from all
10 </Directory>
11 -----
12 # 在/var/www/html/linux-dash/app目录下创建.htaccess文件
13 [root@Lab-06-Task-01 ~]# vi /var/www/html/linux-dash/app/.htaccess
14 ----- .htaccess -----
15 AuthType Basic
16 AuthName "Restricted Files"
17 AuthUserFile /var/www/html/linux-dash/app/.htpasswd
18 Require valid-user
19 -----
20 # 设置访问linux-dash页面的登录账号和密码
21 [root@Lab-06-Task-01 ~]# htpasswd -c /var/www/html/linux-dash/app/.htpa
    sswd admin
22 New password:
23 Re-type new password:
24 Adding password for user admin
25 # 重新服务
26 [root@Lab-06-Task-01 ~]# systemctl restart httpd
```

在Windows本地客户端使用浏览器访问 <http://172.31.0.61/linux-dash/app/>，输入 linux-dash 的账号和密码：admin，能够访问 Linux-dash 监控页面，如图 6-1 所示。

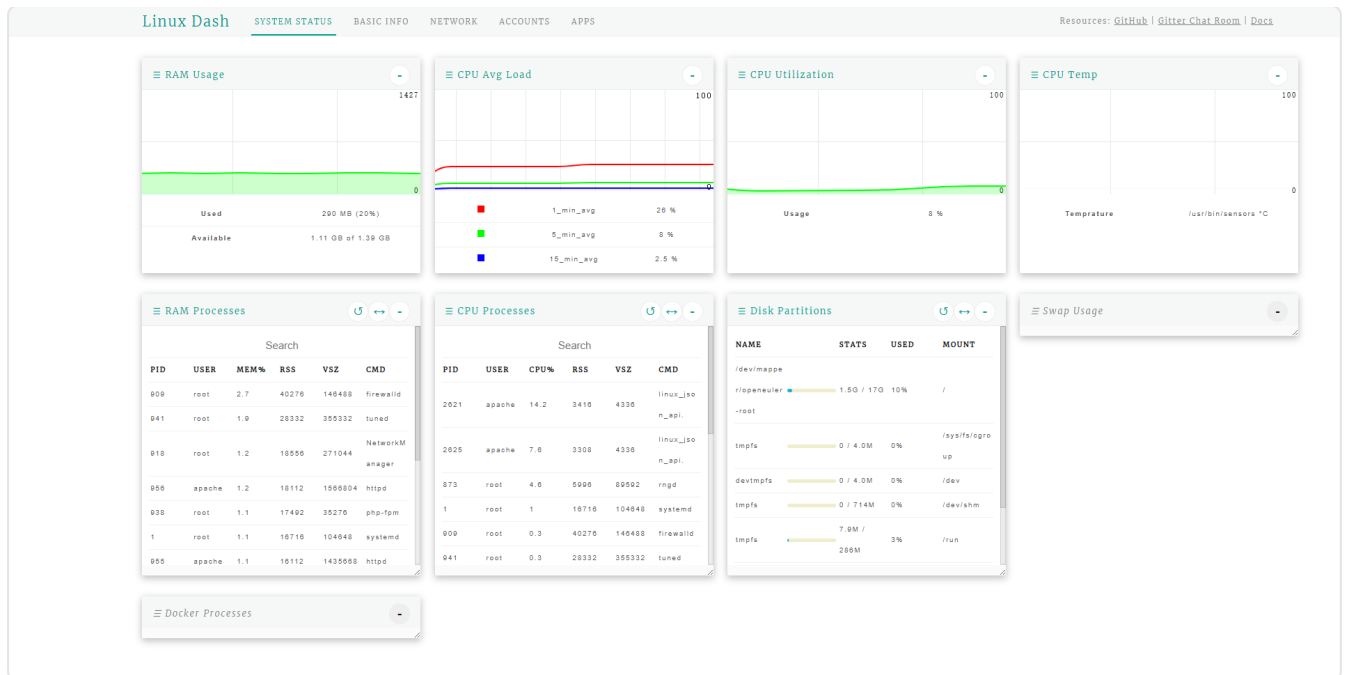


图 6-1 Linux-dash监控

3、使用 SELinux 提升系统的安全性

3.1 配置 SELinux

(1) 管理 SELinux 的工作模式

使用 `getenforce` 命令查看当前的工作模式，使用 `setenforce` 命令在强制模式和宽容模式间进行切换。

Shell

```
1 # 查看当前SELinux的工作模式
2 [root@Lab-06-Task-01 ~]# getenforce
3 # 默认为强制模式
4 Enforcing
5
6 # 修改SELinux的工作模式为宽容模式
7 [root@Lab-06-Task-01 ~]# setenforce 0
8 # 查看修改后模式
9 [root@Lab-06-Task-01 ~]# getenforce
10 Permissive
11
12 # 恢复SELinux的运行模式为强制模式
13 [root@Lab-06-Task-01 ~]# setenforce 1
14 # 查看修改后模式
15 [root@Lab-06-Task-01 ~]# getenforce
16 Enforcing
```

(2) 更改 SELinux 的工作模式和运行状态

永久修改工作模式或者关闭 SELinux，需对 SELinux 的配置文件进行修改，修改完成后重新启动操作系统方可生效。

Shell

```
1 # 查看系统当前SELinux的运行状态
2 [root@Lab-06-Task-01 ~]# cat /etc/selinux/config | grep '^SELINUX='
3 SELINUX=enforcing
4
5 # 修改配置文件实现SELinux为关闭状态
6 [root@Lab-06-Task-01 ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/
  g' /etc/selinux/config
7 # 重启操作系统
8 [root@Lab-06-Task-01 ~]# reboot
9 # 检验状态修改是否生效
10 [root@Lab-06-Task-01 ~]# getenforce
11 Disabled
12 # 修改配置文件实现SELinux为开启状态
13 [root@Lab-06-Task-01 ~]# sed -i 's/SELINUX=disabled/SELINUX=enforcing/
  g' /etc/selinux/config
14 # 重启操作系统
15 [root@Lab-06-Task-01 ~]# reboot
16 # 检验状态修改是否生效
17 [root@Lab-06-Task-01 ~]# getenforce
18 Enforcing
```

3.2 安装 SELinux 管理工具

SELinux 常用的管理工具有 chcon、semange 等，本实验步骤选用 semange 工具。semange 工具集成在 policycoreutils-python-utils 软件中，可使用 yum 工具安装。

Shell

```
1 # 使用yum工具安装policycoreutils-python-utils
2 [root@Lab-06-Task-01 ~]# yum install -y policycoreutils-python-utils
```

3.3 依据场景设计 SELinux

通过 SELinux 提升用户操作的安全性。

需求描述：

第一：修改系统用户映射到 SELinux 内核用户的类型，实现创建用户时 SELinux 用户类型为 user_u。

```

1 # 查看系统默认用户类型
2 [root@Lab-06-Task-01 ~]# semanage login -l
3 登录名                SELinux 用户                MLS/MCS 范围                服务
4 # 系统默认用户的 SELinux 用户类型为 unconfined_u (未限制)
5 __default__            unconfined_u                s0-s0:c0.c1023            *
6 root                   unconfined_u                s0-s0:c0.c1023            *
7
8 # 修改系统默认用户的 SELinux 用户类型
9 [root@Lab-06-Task-01 ~]# semanage login -m -s user_u -r s0 __default__
10
11 # 修改后重新验证查看是否配置成功
12 [root@Lab-06-Task-01 ~]# semanage login -l
13 登录名                SELinux 用户                MLS/MCS 范围                服务
14 # 查看系统默认用户的 SELinux 用户类型已经更改为 user_u (普通用户类型)
15 __default__            user_u                        s0                          *
16 root                   unconfined_u                s0-s0:c0.c1023            *
17
18 # 创建新的用户，并使用新用户进行登录验证
19 [root@Lab-06-Task-01 ~]# adduser testuser
20 #设置密码
21 [root@Lab-06-Task-01 ~]# passwd testuser
22
23 # 切换为新用户进行登录，查看该用户的安全上下文信息
24 [testuser@Lab-06-Task-01 ~]# id -Z
25 user_u:user_r:user_t:s0

```

4、使用防火墙进行系统安全防护

4.1 配置防火墙

(1) 管理防火墙服务

对防火墙服务的管理包括查看防火墙 Firewalld 服务状态、开启、关闭、重启、重新载入防火墙策略等。

Shell

```
1 # 查看防火墙Firewalld服务状态
2 [root@Lab-06-Task-01 ~]# systemctl status firewalld
3
4 # 关闭防火墙服务
5 [root@Lab-06-Task-01 ~]# systemctl stop firewalld
6
7 # 开启防火墙服务
8 [root@Lab-06-Task-01 ~]# systemctl start firewalld
9
10 # 重启防火墙服务
11 [root@Lab-06-Task-01 ~]# systemctl restart firewalld
12
13 # 设置防火墙为开机不自启
14 [root@Lab-06-Task-01 ~]# systemctl disable firewalld
15
16 # 设置防火墙为开机自启动
17 [root@Lab-06-Task-01 ~]# systemctl enable firewalld
18
19 # 重新载入防火墙规则
20 [root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

(2) 配置防火墙日志

对防火墙日志的配置有全局日志配置和规则日志配置两部分。全局日志配置是对防火墙日志规则进行配置，防火墙日志服务由系统 rsyslog 服务进行管理，日志默认存放在 /var/log/firewalld 日志文件中，日志文件基于日期时间自动归档。规则日志配置是设置防火墙触发特定防火墙规则时记录日志的方式。

```
1 # 全局日志配置
2 # 实现防火墙对单播网络通信的日志记录。
3 # 防火墙日志存放目录变更为/var/log/firewalldlog。
4 # 防火墙日志记录等级调整为所有等级的日志均记录。
5 # 使用vi命令编辑/etc/firewalld/firewalld.conf文件
6 [root@Lab-06-Task-01 ~]# vi /etc/firewalld/firewalld.conf
7 # -----/etc/firewalld/firewalld.conf文件-----
8 #
9 # firewallld.conf配置文件内容较多，本部分仅显示与防火墙日志配置有关的内容
10 # 将LogDenied=off改为LogDenied=unicast，实现对单播网络通信的日志记录
11 LogDenied=unicast
12 # -----/etc/firewalld/firewalld.conf文件-----
13 #
14 # 使用vi命令编辑/etc/rsyslog.conf文件
15 [root@Lab-06-Task-01 ~]# vi /etc/rsyslog.conf
16 # -----/etc/rsyslog.conf文件-----
17 # 在配置文件中增加以下内容，kern.*表示记录所有等级的系统内核产生的日志信息
18 kern.* /var/log/firewalldlog/loginfo
19 # -----/etc/rsyslog.conf文件-----
20 #
21 # 创建防火墙日志存放目录
22 [root@Lab-06-Task-01 ~]# mkdir /var/log/firewalldlog
23 # 重新载入配置文件
24 [root@Lab-06-Task-01 ~]# systemctl reload firewalld
25 # 重启日志相关服务
26 [root@Lab-06-Task-01 ~]# systemctl restart rsyslog
27 #
28 # 规则日志配置
29 # 允许本地主机（172.20.1.36）通过httpd服务访问服务器。
30 # 实现触发规则的通信的日志记录。
31 # 设置日志记录的频率为每秒最多3条。
32 # 根据防火墙规则要求配置
33 [root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=172.20.1.36 service name="http" log level=notice prefix="HTTP" limit value="3/s" accept'
34 #
35 # 重新载入防火墙配置使其生效
36 [root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

4.2 依据场景设计防火墙

(1) 通过防火墙指定端口和协议允许访问。

需求描述：

第一：打开 443/TCP 端口。

第二：永久打开 3690/TCP 端口。

第三：永久打开 100-500/TCP 端口（指定范围内端口全部打开）。

Shell

```
1 # 打开443/TCP端口
2 [root@Lab-06-Task-01 ~]# firewall-cmd --add-port=443/tcp
3
4 # 永久打开3690/TCP端口
5 [root@Lab-06-Task-01 ~]# firewall-cmd --add-port=3690/tcp --permanent
6
7 # 永久打开100-500/TCP端口（指定范围内端口全部打开）
8 [root@Lab-06-Task-01 ~]# firewall-cmd --add-port=100-500/tcp --permanen
  t
9
10 # 重新载入防火墙配置
11 [root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

(2) 通过防火墙提升远程连接服务的安全性。

需求描述：

第一：允许地址 172.20.1.36/24 内的客户端远程连接服务器，进行远程管理维护。

第二：客户端远程连接服务器时，每分钟最多允许 5 次远程连接，禁止频繁请求。

Shell

```
1 # 使用firewall-cmd命令删除默认ssh服务规则
2 [root@Lab-06-Task-01 ~]# firewall-cmd --permanent --remove-service=ssh
3
4 # 使用firewall-cmd命令添加指定地址能够远程访问的规则
5 [root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=172.20.1.36/24 service name="ssh" limit value="5/s" accept'
6
7 # 重新载入防火墙配置
8 [root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

(3) 通过防火墙指定 IP 地址允许 / 禁止访问。

需求描述：

第一：允许来自 IP 地址为 172.31.0.11/24 的主机的流量通过防火墙。

第二：禁止来自 IP 地址为 172.31.0.21/24 的主机的流量通过防火墙。

Shell

```
1 # 允许来自IP地址为172.31.0.11/24的主机的流量通过防火墙
2 [root@Lab-13-Task-01 ~]# firewall-cmd --add-source=172.31.0.11 --permanent
3
4 # 禁止来自IP地址为172.31.0.21/24的主机的流量通过防火墙
5 [root@Lab-13-Task-01 ~]# firewall-cmd --remove-source=172.31.0.21 --permanent
6
7 # 重新载入防火墙配置
8 [root@Lab-13-Task-01 ~]# firewall-cmd --reload
```

(4) 通过防火墙提升文件传输服务的安全性。

需求描述：允许地址范围 172.20.1.36/24 内的客户端通过主动与被动模式访问 FTP 服务器。

```
1 # 使用firewall-cmd命令添加通过主动模式访问FTP服务器
2 [root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=172.20.1.36/24 port port=20-21 protocol=tcp limit value="10/m" accept'
3
4 # 使用firewall-cmd命令添加本地客户端允许访问phpMyAdmin
5 [root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=172.20.1.36/24 port port=9000-9020 protocol=tcp limit value="10/m" accept'
6
7 # 重新载入防火墙配置
8 [root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

八、实验考核

实验考核分为【实验随堂查】和【实验线上考】两个部分。

实验随堂查：每个实验设置 5 个考核点。完成实验任务后，按照考核点要求，学生提交实验成果的截图或录屏视频。通过线上考核平台（如课堂派）进行作答。依据提交成果进行评分。

实验线上考：每个实验设置 5 道客观题。通过线上考核平台（如课堂派）进行作答。系统自动评分。

1、实验随堂查

本实验随堂查设置提交实验成果 -5 个截图 / 视频，具体如下：

题目 1[文件题]：提交使用 “mpstat -P ALL 5 2” 命令查看 CPU 信息的截图；

题目 2[文件题]：提交使用 yum 工具成功安装 Apache 的截图；

题目 3[文件题]：提交在 Windows 本地客户端使用浏览器访问 Linux-dash 监控页面的截图；

题目 4[文件题]：提交使用 SELinux 用户类型为 user_u 的账户登录操作系统后，运行 “id -Z” 查看该用户的安全上下文信息的截图；

题目 5[文件题]：提交完成场景设计的防火墙后，运行 “firewall-cmd --list-all” 查看防火墙运行策略的截图；

2、实验线上考

本实验线上考共 5 题。